Available online at www.sciencedirect.com
ScienceDirect
Computers and Mathematics with Applications 53 (2007) 918–926
An International Journal
computers & mathematics with applications
www.elsevier.com/locate/camwa

A fast diagnosis algorithm for locally twisted cube multiprocessor systems under the MM* model

Hui Yang*, Xiaofan Yang

College of Computer Science, Chongqing University, Chongqing, 400044, China

Received 21 April 2006; received in revised form 4 September 2006; accepted 14 September 2006

Abstract

Comparison-based diagnosis is a practical approach to the system-level fault diagnosis of multiprocessors. The locally twisted cube is a newly introduced hypercube variant, which not only possesses lower diameter and better graph embedding capability as compared with a hypercube of the same size, but retains some nice properties of hypercubes. This paper addresses the fault diagnosis of locally twisted cubes under the MM* comparison model. By utilizing the existence of abundant cycles within a locally twisted cube, we present a new diagnosis algorithm. With elaborately organized data, this algorithm can run in $O(N \log_2^2 N)$ time, where N stands for the total number of nodes. In comparison, the classical Sengupta–Dahbura diagnosis algorithm takes as much as $O(N^5)$ time to achieve the same goal. As a consequence, the proposed algorithm is remarkably superior to the Sengupta–Dahbura algorithm in terms of the time overhead.

Keywords: Multiprocessor; System-level fault diagnosis; Comparison model; Diagnosis algorithm; Locally twisted cube

1. Introduction

With the rapid development of technology, the need for high-speed parallel processing systems has been continuously increasing. The reliability of processors in parallel computing systems is therefore becoming an important issue. In order to maintain the reliability of a system, whenever a processor (node) is found faulty, it should be replaced by a fault-free processor. The process of identifying all the faulty nodes is called the diagnosis of the system [1].

Previous studies have proposed various models for diagnosis [2–4]. An important approach, first proposed by Maeng and Malek [2,3] is called the comparison diagnosis model (MM model). The MM model deals with the diagnosis by sending the same input (or task) from a node w to each pair of distinct neighbors, u and v, and then comparing their responses; the result of the comparison indicates an agreement or a disagreement in the two responses. The central task is to identify the faulty nodes by interpreting the comparison results. Sengupta and Dahbura [5] further suggested a modification of the MM model, called the MM* model, in which every processor is assigned to test two

* Corresponding author.
E-mail addresses: yanghui_cqu@126.com, yanghui_cqu@hotmail.com (H. Yang).

0898-1221/$ - see front matter
doi:10.1016/j.camwa.2006.09.008

Table 1
The MM* model

The two tested nodes u and v	The comparator w	$r((u,v)_w)$
Both are fault-free	Fault-free	0
As least one is faulty	Fault-free	1
Any case	Faulty	Unpredictable

other processors whenever it is adjacent to them. Under the MM* model, Sengupta and Dahbura characterized the diagnosable systems and presented an $O(N^5)$ diagnosis algorithm for general diagnosable systems with N processors.

Due to nice properties such as logarithmic numbers of links per node and logarithmic diameter, symmetry, recursive structure, and simple yet efficient communication algorithms, the n-dimensional hypercube enjoys popularity as a topology of interconnection networks [6].

In order to achieve better performance, a variety of hypercube variants were proposed. Among others, the Möbius cube [7], the crossed cube [8], the twisted cube [6,9,10] have diameters of about half of that of a hypercube of the same size. A common feature of these variants is that the labels of some neighboring nodes may differ in a large number of bits. As a result, some of the good properties of hypercubes are lost in these variants. In order to retain as many nice properties of the hypercube as possible, Yang [11] proposed a new hypercube variant known as the *locally twisted cube*, where the labels of any two adjacent nodes differ in at most two successive bits.

An n-dimensional locally twisted cube has the same number of nodes and the same number of edges as an n-dimensional cube, but has half the diameter and better graph embedding capability as compared with its hypercube counterpart [11–14]. It is known that, under the MM* model, all the faulty processors in an n-dimensional locally twisted cube can be identified correctly and completely, provided that the number of faulty processors is bounded by n [15]. As a consequence, the Sengupta–Dahbura's diagnosis algorithm is applicable to locally twisted cube systems, although with a high computational complexity.

Motivated by the cycle decomposition properties used in [16–19], we propose a new diagnosis algorithm tailored for a locally twisted cube connected system under the MM* model. By introducing appropriate data structures, this diagnosis algorithm runs in $O(N \log_2^2 N)$, which is significantly superior to Sengupta–Dahbura's algorithm when applied to locally twisted cube systems.

The rest of the paper is organized in the following way: Preliminaries are provided in Section 2. In Section 3, we expose the cycle decomposition properties of locally twisted cubes. Section 4 is devoted to explanation and formal description of the tailored diagnosis algorithm, with the proof of correctness and the analysis of time complexity. Some concluding remarks are made in Section 5.

2. Preliminaries

The topology of a multiprocessor system is modeled by a graph $G = (V(G), E(G))$, in which each node represents a processor and each edge represents a communication link between two processors. For a node u of G, denote by $N(u)$ the set of all its neighboring nodes, i.e., $N(u) = \{v \in V(G) : v \text{ is adjacent to } u\}$. For a subset S of $V(G)$, let $N(S) = \cup_{v \in S} N(v)$. A Hamiltonian cycle within a graph G is a cycle that passes through each and every node of G exactly once. Let $G[S]$ denote the subgraph of G induced by a set S of nodes. For fundamental graph–theoretic terminology, the reader is referred to Ref. [8].

Diagnosis by the comparison approach can be modeled by a labeled multigraph, called the comparison graph $M(G) = (V(G), C(G))$, where $V(G)$ is the set of all processors and $C(G)$ is the set of labeled edges. A labeled edge $(u, v)_w \in C(G)$, with w being a label on the edge, connects u and v, which implies that processors u and v are being compared by w. For $(u, v)_w \in C(G)$, the output of comparator w of u and v is denoted by $r((u, v)_w)$, a disagreement of the output is denoted by the comparison results $r((u, v)_w) = 1$, whereas an agreement is denoted by $r((u, v)_w) = 0$. The collection of all the comparison results is called a syndrome and denoted by r.

Under the MM* model, each processor w such that $(w, u) \in E(G)$ and $(w, v) \in E(G)$ is a comparator for the pair of processors u and v. The comparison graph $M(G) = (V(G), C(G))$ of a given system can be a multigraph because the same pair of nodes may be compared by several different comparators. The MM* model assumes that a fault-free comparator can give correct comparison results, whereas the outcome of a comparison conducted by a faulty comparator is completely unreliable. More specifically, the assumptions made in MM* are listed in Table 1.

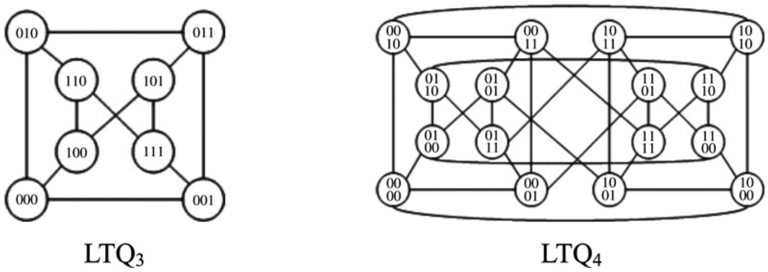


Fig. 1. Two small-sized locally twisted cubes.

Definition 2.1. Given a syndrome r on system G, a cycle in G is *r-zero* if $r\,((u, v)_w) = 0$ for any node w and the two neighboring nodes u, v of w on the cycle. Otherwise the cycle is *r-nonzero*. All nodes on an r-zero cycle are *r-zero*. All nodes on an r-nonzero cycle are *r-nonzero*.

The following result follows from the assumption of the MM* model.

Lemma 2.1. *Let r be a syndrome on a graph G. Let C be a cycle in G.*

(1) *If C is an r-zero cycle, then all nodes on C are of the same status, namely they are either all fault-free or all faulty.*
(2) *If C is an r-nonzero cycle, then at least one node on C is faulty.*

Proof. (1) If the claim was not true, there would be two successive node u and w on C such that u is faulty and w is fault-free. Let v be the other neighbor of w on C. Then $r((u, v)_w) = 1$, a contradiction.
(2) There are three successive nodes, say u, w, and v, on C such that $r((u, v)_w) = 1$. If w is faulty, the result follows. □

Definition 2.2 ([11]). Let $n \geq 2$ be an integer. An n-dimensional locally twisted cube, LTQ_n, is defined recursively as follows:

(1) LTQ_2 is a graph consisting of four nodes labeled with 00, 01, 10, and 11, respectively, connected by four edges (00, 01), (01, 11), (11, 10), and (10, 00).
(2) For $n \geq 3$, LTQ_n is built from two disjoint copies of LTQ_{n-1} according to the following steps: Let $0\mathrm{LTQ}_{n-1}$ denote the graph obtained from one copy of LTQ_{n-1} by prefixing the label of each node with 0. Let $1\mathrm{LTQ}_{n-1}$ denote the graph obtained from the other copy of LTQ_{n-1} by prefixing the label of each node with 1. Connect each node $0x_2x_3\cdots x_n$ of $0\mathrm{LTQ}_{n-1}$ to the node $1(x_2 + x_n)x_3\cdots x_n$ of $1\mathrm{LTQ}_{n-1}$ with an edge.

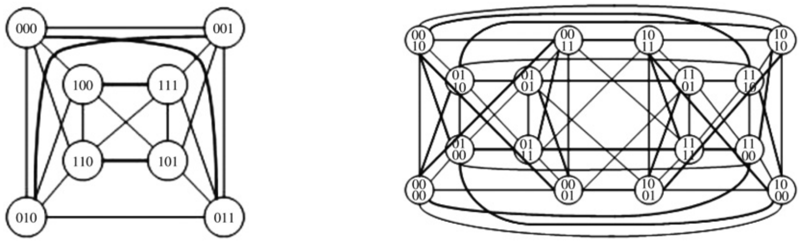
Fig. 1 shows two examples of locally twisted cubes. Let $\{0, 1\}^n$ denote the whole set of all 0-1 binary strings of length n. The locally twisted cubes can also be equivalently defined in the following non-recursive fashion.

Definition 2.1′ ([11]). For $n \geq 2$, the n-dimensional locally twisted cube, LTQ_n, is a graph with $\{0, 1\}^n$ as the node set. Two nodes $x = x_1x_2x_3\cdots x_n$ and $y = y_1y_2y_3\cdots y_n$ of LTQ_n are adjacent if and only if one of the following conditions are satisfied:

(1) There is an integer $1 \leq k \leq n-2$ such that
  (a) $x_k = \bar{y}_k$,
  (b) $x_{k+1} = y_{k+1} + x_n$, and
  (c) all the remaining bits of x and y are identical.
(2) There is an integer $k \in \{n-1, n\}$ such that x and y differ only in the kth bit.

If so, then y is called the k-th dimensional neighbor of x, and vice versa.

Lemma 2.2 ([12]). $\mathrm{LTQ}_n (n \geq 2)$ *contains a Hamiltonian cycle (which can be constructed recursively).*



$ELTQ_3$ $ELTQ_4$

Fig. 2. Two small-sized enhanced locally twisted cubes.

3. Cycle decomposition properties of locally twisted cube

This section aims at revealing the cycle decomposition properties of locally twisted cubes. For two 0–1 binary strings x and y, let xy denote the concatenation of x and y.

For integer $n \geq 5$, let $c(n) = \lceil \log_2(n+1) \rceil$, then $c(n) \geq 3$. For any given $x \in \{0,1\}^{n-c(n)}$, let $V(x) = \{xy : y \in \{0,1\}^{c(n)}\}$, then $V(x)$ is a set of $2^{c(n)}$ nodes of LTQ_n, where each node has a label prefixed with x. Clearly, the induced subgraph $LTQ_n[V(x)]$ is isomorphic to $LTQ_{c(n)}$. Therefore, $LTQ_n (n \geq 5)$ can be decomposed into a family $\{LTQ_n[V(x)] : x \in \{0,1\}^{n-c(n)}\}$ of $2^{n-c(n)}$ disjoint subgraphs, each of which is isomorphic to $LTQ_{c(n)}$. In what follows, we use the symbol $LTQ_n(x)$ to denote $LTQ_n[V(x)]$.

Let HC be a Hamiltonian cycle within $LTQ_{c(n)}$. For any given $x \in \{0,1\}^{n-c(n)}$, we define a mapping $f_x : V(LTQ_{c(n)}) \to V(x)$ such that $f_x(y) = xy$ for each $y \in \{0,1\}^{c(n)}$. Then HC is mapped onto a Hamiltonian cycle of $LTQ_n(x)$, which is also a cycle within LTQ_n. We call this cycle a cycle induced by HC with respect to x and denote it by HC(x). Thus, LTQ_n contains a set of $2^{n-c(n)}$ disjoint cycles of length $2^{c(n)}$. On this basis, we introduce the following notions:

Definition 3.1. For any integer $n \geq 5$. Let $c(n) = \lceil \log_2(n+1) \rceil$. Let HC be a Hamiltonian cycle in $LTQ_{c(n)}$. The set of all cycles of LTQ_n induced by HC, denoted by CD(HC), is called a cycle decomposition of LTQ_n induced by HC, i.e. CD(HC) $= \{HC(x) : x \in \{0,1\}^{n-c(n)}\}$. Two cycles in CD(HC) are adjacent if and only if there is a node on one cycle that is adjacent to some node on the other cycle.

Depending on the choice of the Hamiltonian cycle HC within $LTQ_{c(n)}$, there are a large number of different cycle decompositions for LTQ_n. Therefore, HC is called the base cycle that induces the cycle decomposition. For our purpose, any cycle decomposition is appropriate.

We now construct a graph $T = (V(T), E(T))$ from LTQ_n by contracting each subgraph $LTQ_n(x)$ to a single node with label x. That is, $V(T) = \{0,1\}^{n-c(n)}$. Two distinct nodes x and y of T are adjacent if and only if $LTQ_n(x)$ has a node that is adjacent to some node of $LTQ_n(y)$.

In order to discuss the topological properties of the graph T, we introduce a locally twisted cube variant called an n-dimensional enhanced locally twisted cube for it is obtained by adding $(n-1) \cdot 2^{n-1}$ more edges to a locally twisted cube of 2^n nodes.

Definition 3.2. For $n \geq 2$, an n-dimensional enhanced locally twisted cube, $ELTQ_n$, is constructed from an n-dimensional locally twisted cube LTQ_n by adding $(n-1) \cdot 2^{n-1}$ more edges. Two nodes $x = x_1 x_2 \cdots x_n$ and $y = y_1 y_2 \cdots y_n$ of $ELTQ_n$ are connected by an extra edge if and only if one of the following conditions is satisfied:

(1) If $x_n = 0$, and there is an integer $1 \leq k \leq n-1$ such that $y_k = \bar{x}_k$, and $y_{k+1} = x_{k+1} + \bar{x}_n$, and $x_r = y_r$ for all the remaining bits.
(2) If $x_n = 1$, and there is an integer $1 \leq k \leq n-2$ such that $y_k = \bar{x}_k$, and $y_{k+1} = x_{k+1} + \bar{x}_n$, and $x_r = y_r$ for all the remaining bits.
(3) If $x_n = 1$, and there is an integer $k = n-1$ such that $y_k = \bar{x}_k$, and $y_{k+1} = \bar{x}_n$, and $x_r = y_r$ for all the remaining bits.

From the definition of enhanced locally twisted cubes, every node of $ELTQ_n$ has $2n-1$ neighboring nodes. Fig. 2 shows two small-sized $ELTQ_n$.

By observing the topology of the enhanced locally twisted cubes, an ELTQ_{n+1} can be constructed from two disjoint copies of ELTQ_n according to the following steps: Let $0\mathrm{ELTQ}_n$ denote the graph obtained from one copy of ELTQ_n by preceding the label of each node with 0. Let $1\mathrm{ELTQ}_n$ denote the graph obtained from the other copy of ELTQ_n by preceding the label of each node with 1. Connect each node $0x_1x_2\cdots x_n$ of $0\mathrm{ELTQ}_n$ to the nodes $1x_1x_2\cdots x_n$ and $1\bar{x}_1x_2\cdots x_n$ of $1\mathrm{ELTQ}_n$ with an edge, respectively.

Lemma 3.1. *For any two nodes u, v in* ELTQ_n, $|N(u,v)| \geq 4n-6$.

Proof. We prove the lemma by induction on n, the dimension of the enhanced locally twisted cube.

Basis. When $n = 2$ or 3, the claim can be checked by inspection.

Hypothesis. The claim holds for ELTQ_n.

Induction. Consider an $(n+1)$-dimensional enhanced locally twisted cube, ELTQ_{n+1}. An ELTQ_{n+1} is composed of two n-dimensional enhanced locally twisted cubes $0\mathrm{ELTQ}_n$ and $1\mathrm{ELTQ}_n$ such that each node $0x_1x_2\cdots x_n$ in $0\mathrm{ELTQ}_n$ is adjacent to two nodes $1x_1x_2\cdots x_n$ and $1\bar{x}_1x_2\cdots x_n$ in $1\mathrm{ELTQ}_n$. There are three possibilities.

(1) If u and v fall in different ELTQ_n, since each node in ELTQ_n has $2n-1$ neighbors. Therefore,

$$|N(u,v)| \geq 2*(2n-1) = 4n-2 = 4(n+1)-6.$$

(2) If u and v fall in the same ELTQ_n. Without loss of generality, suppose both u and v fall in $0\mathrm{ELTQ}_n$. By the topological properties of enhanced twisted cubes, only if u and v are adjacent and their labels are $00x_2\cdots x_n$ and $01x_2\cdots x_n$, have they the same neighbors in $1\mathrm{ELTQ}_n$, which are labeled $10x_2\cdots x_n$ and $11x_2\cdots x_n$. When this is the case, u and v have at most two common neighbors in $0\mathrm{ELTQ}_n$, which are labeled $00\bar{x}_2\cdots x_n$ and $01\bar{x}_2\cdots x_n$. Therefore,

$$|N(u,v)| \geq 2*(2n-1)-2+2 = 4n-2 = 4(n+1)-6.$$

(3) If u and v fall in the same ELTQ_n, and their labels are different from those in the case (2), then, by hypothesis, u and v have at least $4n-6$ neighbors, all in the same ELTQ_n. But u and v have four more neighbors in the other ELTQ_n. Therefore,

$$|N(u,v)| \geq (4n-6)+4 = 4(n+1)-6. \quad \square$$

Lemma 3.2. *T is isomorphic to* $\mathrm{ELTQ}_{n-c(n)}$.

Proof. Assume two distinct nodes $x = x_1x_2\cdots x_{n-c(n)}$ and $y = y_1y_2\cdots y_{n-c(n)}$ of T are adjacent. Then $\mathrm{LTQ}_n(x)$ has a node $xx_{n-c(n)+1}\cdots x_n$ that is adjacent to some node $yy_{n-c(n)+1}\cdots y_n$ of $\mathrm{LTQ}_n(y)$. Because $x \neq y$, there is $1 \leq k \leq n-c(n)$ such that $y_k = \bar{x}_k$, $y_{k+1} = x_{k+1} + x_n$ and all the remaining bits of $xx_{n-c(n)+1}\cdots x_n$ and $yy_{n-c(n)+1}\cdots y_n$ are identical. Notice that for the same prefix x, the bit x_n is either 0 or 1. Therefore, there are some more edges linked to the node x of T, which are the very extra edges added in $\mathrm{ELTQ}_{n-c(n)}$. Consequently, from the definition of enhanced twisted cubes, x and y are adjacent in $\mathrm{ELTQ}_{n-c(n)}$. Similarly, we can prove that, if x and y are adjacent in $\mathrm{ELTQ}_{n-c(n)}$, then they are also adjacent in T. $\square$

Lemma 3.3. *Every cycle in a cycle decomposition of* LTQ_n *is adjacent to exactly* $2[n-c(n)]-1$ *cycles in the same cycle decomposition.*

Proof. Notice that there is an one-to-one correspondence between the cycles in a cycle decomposition of LTQ_n and the nodes of T. Furthermore, two cycles in the cycle decomposition are adjacent if and only if the two corresponding nodes of T are adjacent. The claim follows from Lemma 3.2. $\square$

Lemma 3.4. *Given two cycles in a cycle decomposition of* LTQ_n*, there are at least* $4[n-c(n)]-6$ *other cycles in this cycle decomposition such that each of them is adjacent to one of the two cycles.*

Proof. There is an one-to-one correspondence between the cycles in a cycle decomposition of LTQ_n and the nodes of T. The claim follows from Lemmas 3.2 and 3.1. $\square$

4. A new diagnosis algorithm

In order to describe the new diagnosis algorithm, we need the following notions.

Definition 4.1. Let r be a syndrome on LTQ_n. Let C be a cycle in a cycle decomposition CD(HC) of LTQ_n.

(1) C is r-guarded if C is r-nonzero but is adjacent to some r-zero cycle. All the nodes on an r-guarded cycle are r-guarded.
(2) C is r-unguarded if C is r-nonzero and is not adjacent to any r-zero cycle. All the nodes on an r-unguarded cycle are r-unguarded.

Our diagnosis algorithm is based on the following theorem.

Theorem 4.1. *Let $n \geq 7$. Assume there are at most n faulty nodes in* LTQ_n. *Let* CD(HC) *be a cycle decomposition of* LTQ_n. *Let r be a syndrome on* LTQ_n *generated by the fault set.*

(1) *Every r-zero cycle in* CD(HC) *consists of fault-free nodes.*
(2) *There is an r-zero cycle in* CD(HC).
(3) *There is at most one r-unguarded cycle in* CD(HC).
(4) *If there is an r-unguarded cycle in* CD(HC), *then there is at most one node on this cycle such that all of its r-guarded neighboring nodes are faulty.*
(5) *If there is an r-unguarded cycle in* CD(HC), *and there is one node on this cycle such that* (a) *all of its r-guarded neighboring nodes are faulty, and* (b) *each of its r-unguarded neighboring nodes either is faulty or is not adjacent to any fault-free r-guarded nodes. Then the node is fault-free.*

Proof. (1) Every cycle in CD(HC) consists of $2^{c(n)} = 2^{\lceil \log_2(n+1) \rceil} (\geq n+1)$ nodes. The claim follows from Lemma 2.1.
(2) There are totally $2^{n-c(n)} = 2^{n-\lceil \log_2(n+1) \rceil} (\geq n+1)$ cycles in CD(HC). It follows from Lemma 2.1 that there is a fault-free cycle in CD(HC), which must be r-zero.
(3) Assume there were two distinct r-unguarded cycles, say C_1 and C_2, in CD(HC). According to Lemma 3.4, CD(HC) would contain at least $4[n - c(n)] - 6 = 4[n - \lceil \log_2(n+1) \rceil] - 6 (\geq n+1)$ r-nonzero cycles each of which is either adjacent to C_1 or adjacent to C_2. According to Lemma 2.1, there would be more than n faulty nodes, a contradiction.
(4) Assume there were two distinct nodes, u and v, on an r-unguarded cycle such that all the r-guarded neighboring nodes of u and v are faulty. Then there would be at least $4[n - c(n)] - 2 = 4[n - \lceil \log_2(n+1) \rceil] - 2 (\geq n+1)$ faulty nodes, a contradiction.
(5) Otherwise there would be more than n faulty nodes. □

4.1. Description of algorithm

The main idea of the algorithm is described as follows.

Algorithm Diagnosis

INPUT: An integer $n \geq 7$, a cycle decomposition CD(HC) of LTQ_n, and a syndrome r on LTQ_n.

OUTPUT: A set of nodes which are diagnosed as faulty.

Step 1. Determine all the r-zero and r-nonzero cycles in CD(HC). Determine all the r-zero and r-nonzero nodes. Diagnose all the r-zero nodes as fault-free.

Step 2. For each r-guarded node p, if there is an r-zero node q that is adjacent to p, and an r-zero node w that is adjacent to q, then diagnose p as fault-free or faulty according as $r((p, w)_q) = 0$ or 1.

Step 3. For each r-unguarded node p, if there is an r-guarded node q that is adjacent to p, and a node w that is adjacent to q such that both q and w have been diagnosed as fault-free, then diagnose p as fault-free or faulty according as $r((p, w)_q) = 0$ or 1.

Step 4. For each remaining r-unguarded node p, if there is an r-unguarded node q that is adjacent to p, and an r-guarded node w that is adjacent to q such that both q and w have been diagnosed as fault-free in the previous steps, then diagnose p as fault-free or faulty according as $r((p, w)_q) = 0$ or 1.

Step 5. Let F be the set of all the nodes having been diagnosed as faulty. Return F.

The following theorem, which is a corollary of Theorem 4.1, ensures the correctness of algorithm Diagnosis.

Theorem 4.2. *Let $n \geq 7$. Assume there are at most n faulty nodes in* LTQ_n. *Then algorithm Diagnosis when running on a syndrome produced by a fault set returns the fault set.*

4.2. Design of data structure

In order for algorithm Diagnosis to achieve optimal performance, we need to introduce appropriate data structures, which are described below:

(1) A three-dimensional array **SYNDROME** of size $2^n \cdot n \cdot n$, which records a syndrome r on LTQ_n. The first index of array SYNDROME takes on values in $\{0, 1\}^n$, while the second and third indexes are integers between 1 and n, inclusive. For $x \in \{0, 1\}^n$, $1 \leq i \leq n$, $1 \leq j \leq n(j \neq i)$, SYNDROME$[x][i][j]$ stores the value of $r((y, z)_x)$, where y and z are the ith dimensional and jth dimensional neighbors of x, respectively.
(2) An one-dimensional array **BCYCLE** of length $2^{c(n)}$, which records a base cycle on $\mathrm{LTQ}_{c(n)}$, i.e. $HC : (y^1 \rightarrow y^2 \rightarrow \cdots \rightarrow y^{2^{c(n)}} \rightarrow y^1)$. For each index $1 \leq i \leq 2^{c(n)}$, BCYCLE $[i] = y^i$.
(3) An one-dimensional array **ZCYCLE** of length $2^{n-c(n)}$ and with indexes in $\{0, 1\}^{n-c(n)}$, which is used to distinguish r-zero cycles from r-nonzero cycles. For each $x \in \{0, 1\}^{n-c(n)}$, ZCYCLE$[x]$ equals 0 or 1 according as cycle $HC(x)$ is r-zero or not. Initially, all the components of ZCYCLE are set as 0.
(4) An one-dimensional array **DIAG** of length 2^n and with indexes in $\{0, 1\}^n$, which records the diagnosis result. For $x \in \{0, 1\}^n$, DIAG$[x]$ stores the diagnosis result about x, which assumes a value in the set $\{-1, 0, 1\}$. The three values denote three states of a node, that is, undiagnosed, fault-free and faulty, respectively. Initially, all the components of DIAG are set as -1.

Based on these data structures, our diagnosis algorithm can be formulated as follows.

Algorithm DIAGNOSIS

INPUT: An integer $n \geq 7$, a three-dimensional array SYNDROME, an one-dimensional array BCYCLE, an one-dimensional array ZCYCLE, an one-dimensional array DIAG, which are described as above.

OUTPUT: A set F of nodes.

/* Initialization */
1. for each $x \in \{0, 1\}^n$ DIAG $[x] = -1$;
2. for each $x \in \{0, 1\}^{n-c(n)}$, ZCYCLE$[x] = 0$;
/* Get all the r-zero cycles and r-nonzero cycles */
3. for each $x \in \{0, 1\}^{n-c(n)}$ for each $i = 1$ to $2^{c(n)}$
4. $\quad$ $p =$ bit position where BCYCLE$[i] \neq$ BCYCLE$[i - 1]$;
5. $\quad$ $q =$ bit position where BCYCLE$[i] \neq$ BCYCLE$[i + 1]$;
6. if (SYNDROME$[x$BCYCLE$[i]][p][q] == 1$) then ZCYCLE$[x] = 1$; break ;
/* Diagnose all the r-zero nodes as fault-free */
7. for each $x \in \{0, 1\}^{n-c(n)}$ if (ZCYCLE$[x] == 0$) then for each $y \in \{0, 1\}^{c(n)}$
$\quad$ DIAG$[xy] = 0$;

/* The remaining part of the algorithm consists of three rounds. The first round identifies all the r-guarded nodes. The second round identifies all those r-unguarded nodes each of which has a fault-free r-guarded neighbor. The third round identifies all those r-unguarded nodes each of which has no fault-free r-guarded neighbor but has a fault-free r-unguarded neighbor that, in turn, is adjacent to a fault-free r-guarded node */

8. for $i = 1$ to 3
9. $\quad$ for each $x \in \{0, 1\}^n$
10. $\quad\quad$ if (DIAG $[x] == 1$) then
11. $\quad\quad\quad$ for $p = 1$ to n
12. $\quad\quad\quad\quad$ $y =$ the pth dimensional neighbor of x;
13. $\quad\quad\quad\quad$ if (DIAG$[y] == 0$) then
14. $\quad\quad\quad\quad$ for $q = 1$ to n except p
15. $\quad\quad\quad\quad\quad$ $z =$ the qth dimensional neighbor of y;

```
16.            if (DIAG[z] == 0) then
17.            if (SYNDROME[y][p][q] == 0) then DIAG[x] = 0 ; goto 9 ;
18.            else DIAG[x] = 1 ; goto 9 ;
/* Diagnose the remaining undiagnosed node as fault-free, if it is existent */
19. for each x ∈ {0, 1}^n if (DIAG [x] == 1) then DIAG[x] = 0 ;
20. F = empty set ;
21. for each x ∈ {0, 1}^n if (DIAG[x] == 1) then F = F ∪ {x};
22. OUTPUT (F).
```

Theorem 4.3. *Algorithm DIAGNOSIS runs in $O(N \log_2^2 N)$, where $N = 2^n$ is the number of nodes of* LTQ_n.

Proof. Statements 1–2 of the algorithm spend $O(2^n)$ time. Statements 3–6 cost $O(n2^n)$ time. Statement 7 runs in $O(2^n)$ time. Statements 8–18 are executed in $O(n^2 2^n)$ time. Statement 19 works in $O(2^n)$ time. Statements 20–22 need $O(2^n)$ time. So the total time needed by the algorithm when running on a syndrome on LTQ_n is $O(n^2 2^n) = O(N \log_2^2 N)$. Note that the size of input for algorithm DIAGNOSIS is $O(n^2 2^n)$. Therefore, the time of DIAGNOSIS is linear in the size of input.

When $n \leq 5$, algorithm DIAGNOSIS may not work properly because there may be no r-zero cycle at all. Furthermore, when $n \leq 6$, DIAGNOSIS may not work properly because there may be more than one r-unguarded cycle. In that case, we may recall Sengupta–Dahbura's algorithm instead to identify the faulty nodes. The combined algorithm, however, also runs in $O(N \log_2^2 N)$ time. □

5. Conclusions

Under the MM* comparison model, we have proposed a diagnosis algorithm tailored for the n-dimensional locally twisted cube system. The correctness of the algorithm has been proved for $n \geq 7$. Based on elaborately designed data structures, the algorithm DIAGNOSIS runs in $O(N \log_2^2 N)$ time. In comparison, Sengupta–Dahbura's diagnosis algorithm runs in $O(N^5)$ time. Therefore, our diagnosis algorithm is remarkably superior.

Acknowledgements

The authors are grateful to the anonymous reviewers for their valuable suggestions. This work has been supported by New Century's Talents Plan Funds of Education Ministry of China (No. NCET-05-0759), Doctorate Funds of Education Ministry of China (No. 200506110001) and Natural Science Funds of Chongqing (No. CSTC2005, BB2191).

References

[1] A.K. Somani, System level diagnosis: A review, Technical Report, Dependable Computing Laboratory, Iowa State University, 1997.
[2] J. Maeng, M. Malek, A comparison connection assignment for self-diagnosis of multiprocessor systems, in: Proc. 11th Int'l Symp. Fault-Tolerant Computing, 1981, pp. 173–175.
[3] M. Malek, A comparison connection assignment for diagnosis of multiprocessor systems, in: Proc. 7th Int'l Symp. Computer Architecture, 1980, pp. 31–35.
[4] F. Preparata, G. Metze, R. Chien, On the connection assignment problem of diagnosable systems, IEEE Trans. Electron. Comput. EC-16 (6) (1967) 848–854.
[5] A. Sengupta, A.T. Dahbura, On self-diagnosable multiprocessor systems: Diagnosis by the comparison approach, IEEE Trans. Comput. 41 (11) (1992) 1386–1396.
[6] F.B. Chedid, R.B. Chedid, A new variation on hypercubes with smaller diameter, Inform. Process. Lett. 46 (6) (1993) 275–280.
[7] P. Cull, S.M. Larson, The Möbius cubes, IEEE Trans. Comput. 44 (5) (1995) 647–659.
[8] F. Harary, Graph Theory, Addison-Wesley, Reading, 1969.
[9] A.-H. Estafahanian, L.M. Ni, B. Sagan, The twisted N-cube with application to multiprocessing, IEEE Trans. Comput. 40 (1) (1991) 88–93.
[10] D.W. Hillis, The connection machine (computer architecture for the new wave), Technical Report, MIT AI Memo 646, Massachusetts Institute of Technology, 1981.
[11] X.F. Yang, D.J. Evans, G.M. Megson, The locally twisted cubes, Int. J. Comput. Math. 82 (4) (2005) 401–413.
[12] X.F. Yang, D.J. Evans, G.M. Megson, Locally twisted cubes are 4-pancyclic, Appl. Math. Lett. 17 (8) (2004) 919–925.
[13] M.J. Ma, J.M. Xu, Panconnectivity of locally twisted cubes, Appl. Math. Lett. 19 (7) (2006) 673–677.
[14] Q.Y. Chang, M.J. Ma, J.M. Xu, Fault-tolerant pancyclicity of locally twisted cubes, J. China Univ. Sci. Tech. 36 (6) (2006) 607–610, 673 (in Chinese).

[15] G.Y. Chang, G.J. Chang, G.H. Chen, Diagnosabilities of regular networks, IEEE Trans. Parallel Distrib. Syst. 16 (4) (2005) 314–323.
[16] S. Khanna, W.K. Fuchs, A graph partitioning approach to sequential diagnosis, IEEE Trans. Comput. 46 (1) (1997) 39–47.
[17] E. Kranakis, A. Pelc, Better adaptive diagnosis of hypercubes, IEEE Trans. Comput. 49 (10) (2000) 1013–1020.
[18] X.F. Yang, A fast pessimistic one-step diagnosis algorithm for hypercube multicomputer systems, J. Parallel Distrib. Comput. 64 (4) (2004) 546–553.
[19] X.F. Yang, G.M. Megson, D.J. Evans, A comparison-based diagnosis algorithm tailored for crossed cube multiprocessor systems, Micpro. Micsyst. 29 (4) (2005) 169–175.